

01 Sep 2023

QC-SANE: Robust Control in DRL using Quantile Critic with Spiking Actor and Normalized Ensemble

Surbhi Gupta


Gaurav Singal

Deepak Garg

Sarangapani Jagannathan

Missouri University of Science and Technology, sarangap@mst.edu

Follow this and additional works at: https://scholarsmine.mst.edu/ele_comeng_facwork

 Part of the [Computer Sciences Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

S. Gupta et al., "QC-SANE: Robust Control in DRL using Quantile Critic with Spiking Actor and Normalized Ensemble," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 6656 - 6662, Institute of Electrical and Electronics Engineers, Sep 2023.

The definitive version is available at <https://doi.org/10.1109/TNNLS.2021.3129525>

This Article - Journal is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

QC_SANE: Robust Control in DRL Using Quantile Critic With Spiking Actor and Normalized Ensemble

Surbhi Gupta^{1b}, Gaurav Singal^{1b}, *Senior Member, IEEE*, Deepak Garg, *Senior Member, IEEE*,
and Sarangapani Jagannathan^{1b}, *Fellow, IEEE*

Abstract—Recently introduced deep reinforcement learning (DRL) techniques in discrete-time have resulted in significant advances in online games, robotics, and so on. Inspired from recent developments, we have proposed an approach referred to as Quantile Critic with Spiking Actor and Normalized Ensemble (QC_SANE) for continuous control problems, which uses quantile loss to train critic and a spiking neural network (NN) to train an ensemble of actors. The NN does an internal normalization using a scaled exponential linear unit (SELU) activation function and ensures robustness. The empirical study on multijoint dynamics with contact (MuJoCo)-based environments shows improved training and test results than the state-of-the-art approach: population coded spiking actor network (PopSAN).

Index Terms—Actor critic, deep reinforcement learning (DRL), ensemble, reinforcement learning (RL), robust control, spiking neural network (SNN).

I. INTRODUCTION

Reinforcement learning (RL) has been extended to deep RL (DRL) for approximation of high-dimensional state or action space problems [1]. In this regard, researchers had proposed many approaches to cope up with the problem encountered in the DRL version of the RL approach [2]–[6] that serves as the baseline for further advances [7], [8]. These approaches jump-started the research in various directions from the core aspect to the application domain. The ongoing research enhances the impact of DRL approaches by incorporating scalability, energy efficiency, generalization, and robustness.

Recently, Patel *et al.* [9] have shown the robustness of the spiking neural network (SNN) to input perturbation by converting a deep Q -network (DQN) to SNN that enables energy-efficient implementation on neuromorphic processors. Tang *et al.* [10] have increased the representation capacity of the SNN using a population coding scheme for continuous control tasks. Klambauer *et al.* [11] incorporated self-normalizing property in neural networks (NNs) using scaled exponential linear unit (SELU) activation function that shows robustness to perturbations.

On the other side, Tagasovska and Lopez-Paz [12] used quantile regression to estimate aleatoric uncertainty. Kuznetsov *et al.* [13] had considered quantile critics to avoid overestimation bias. Chung *et al.* [14] have considered the quantile method for robustness by uncertainty quantification. Since the optimization of a single quantile level may be accurate, it will result in a miscalibrated model.

Manuscript received 15 June 2021; revised 26 September 2021; accepted 16 November 2021. Date of publication 7 December 2021; date of current version 1 September 2023. (*Corresponding author: Gaurav Singal.*)

Surbhi Gupta and Deepak Garg are with the Department of Computer Science Engineering, Bennett University, Greater Noida, Uttar Pradesh 201310, India (e-mail: gsurbhi1993@gmail.com; deepakgarg108@gmail.com).

Gaurav Singal is with the Department of Computer Science Engineering, Netaji Subhas University of Technology, New Delhi 110078, India (e-mail: gaurav.singal@nsut.ac.in; gauravsingal789@gmail.com).

Sarangapani Jagannathan is with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sarangap@mst.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TNNLS.2021.3129525>.

Digital Object Identifier 10.1109/TNNLS.2021.3129525

2162-237X © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

Hence, instead of considering a single quantile value, they have learned quantile simultaneously with conditional density estimates.

Another way of making the learning process robust is by using the ensemble approach as explored in [15]–[17]. Variants of DQN tackled in past research were combined in Rainbow [18] for discrete action settings. Inspired from [18], we propose a robust approach for continuous control tasks that are common in robotics. Due to continuous state-action space, an agent must explore many possible actions and need more samples. The direct translation of DRL approaches from discrete to continuous control setting is not viable as the output layer of the considered NN needs to be designed accordingly [19]. For discrete output, we can estimate the best, but, for continuous output, it is not possible to take the best value among all actions to estimate the target for the critic [2]. We have inherited the ideas to deal with continuous control from the previous approaches [20] that design the output layer to predict the parameters, such as mean and log standard deviation. We use the predicted values for estimating the action with the Gaussian function. Actor updates itself by moving in the direction of gradient of Critic's evaluated value [21].

We aim to predict the critic's quantile values for the policy evaluation as it will quantify the uncertainty in predicting state and action values. Instead of taking either a single or all quantile levels, the proposed work considers three levels of quantile values that allow multiple pinball values and generalizes across them. This brief incorporates the quantile and ensemble of actors with SNN in continuous control setting since SNN is more energy-efficient than feedforward NN. Second, we have shown the theoretical convergence of the policy evaluation for multiple quantile value assessments. The theoretical analysis confirms that the proposed approach will maintain the convergence property.

Unlike the common practice of using a single actor in actor-critic settings, we consider an ensemble of actors in the form of agents through SNN to generate distinct policies. The most common NN used for DRL nonvisionary tasks is a fully connected NN (FNN), and it is used commonly with the batch normalization (deep deterministic policy gradient (DDPG) [2]) technique. FNN trained with normalization is perturbed by several parameters, such as stochastic gradient descent, stochastic regularization, and high variance. The SELU activation function, on the other hand, implicitly does batch normalization, shows robustness to perturbation, and reduces variance in error during training FNN. Hence, finally, our approach performs normalization using SELU nonlinearity.

For the robustness analysis, we have estimated the results on the noisy state as noise can create perturbations, and SNN must be insensitive to it. To corrupt the state, we injected noise in the dimension of the vector forming the state as it resembles sensor value perturbation. As per the author's best knowledge, we are not aware of any effort that integrates quantile critic with the ensemble of SNN-based actors and does internal normalization of NN to manage aleatory uncertainty. The following points illustrate the contributions of the brief.

- 1) For the first time, we have incorporated the uncertainty estimates by predicting quantile values at the critic and employed an ensemble of actors to serve as agents in order to generate

distinct policies with SNN-based actor and make soft-actor-critic (SAC), which considers continuous actions, energy-efficient, and robust.

- 2) The critic network predicts Q -values, and the actor network undergoes internal normalization using SELU activation to aid in stabilizing the variance and robust learning.
- 3) Theoretical convergence of the quantile critic-based policy evaluation confirms that our method will maintain the convergence of the baseline approach.
- 4) We have conducted experiments by adding noise in the dimension of the state, which may happen in the realistic setting while deploying the agent. The empirical results of the proposed Quantile Critic with Spiking Actor and Normalized Ensemble (QC_SANE) show stable learning and robust selection in the noisy scenario.

The brief is divided into five sections, where Section I introduces the idea and Section II provides the detail of the concepts that served as the ingredients for the proposed QC_SANE approach. Section III gives the details of the proposed work and illustrates the detail on how the ingredients are integrated. Section IV provides the experimental details with results and discussion, and Section V concludes the findings with the future direction.

II. BACKGROUND

This section introduces the well-known terminology and architectures that help readers increase their understanding of the proposed work.

A. Spiking NN

Low-dimensional control using artificial NN (ANN) does not provide energy-efficient and robust control as these networks lack internal temporal coding [22]. The first generation ANN (such as perceptron network, Boltzmann machine, and hop-field network) produces digital output based on the threshold as these neurons consider linear activation, and the neurons of second-generation ANN [such as convolution NN (CNN) and recurrent NN (RNN)] can generate continuous output by applying nonlinear activation. The neurons of the next (third) generation ANN aka SNN are known as spiking neurons as these neurons more closely model the functionality of the biological neurons.

In these neurons, spikes are fired only when the membrane potential increases a threshold. Among various types of spiking neurons, such as integrate-and-fire (IF), subtractive IF (SubIF), leaky IF (LIF) neuron, and stochastic LIF neuron, we have used LIF neuron in SNN. The capabilities of SNN had been explored in supervised learning tasks [22] and RL task of Atari games [9], continuous control manipulation [23], and locomotion [10] task. Population coding technique is used in [10] to increase the representation capacity of SNNs as the conversion of deep NN (DNN) to SNN results in inferior performance [24] and SNN with rate coding representation technique still suffers in the high-dimensional complex task and requires more precise encoding. Population coding used with SNN encodes each dimension of the observation and action space to increase the representation capability of spiking neurons. The advanced hardware chips, such as Intel's Loihi and IBM's TrueNorth, support machine learning with energy-efficient implementation by employing SNN [9]. Though the concept of spiking neuron is not new, it gains acceleration as the research progress [25].

B. Quantile Regression

Unlike mean absolute error (MAE) and mean square error (MSE), quantile loss considers the general case. MAE loss can well manage

outliers than MSE loss that further increases the error by taking square. Despite robust estimation of MAE, the commonly chosen loss in DRL is the MSE loss as its gradient is variable and proportional to the error, while MAE loss has a constant gradient everywhere except at the zero error point. The quantile loss penalizes positive and negative errors based on the chosen quantile. Hence, quantile regression penalizes underestimation and overestimation. The quantile regression considers the prediction at specified quantile. Instead of predicting a single value, it is based on the prediction interval that is well-calibrated. It considers uncertainty in the predicted value by capturing aleatoric uncertainty that arises due to the measurement error or latent variables and helps in noisy structure analysis [12].

Quantile regression leads to robust estimation depending on the level of quantile considered. For example, 0.75 level of quantile penalizes more on under prediction, while 0.25 level of quantile penalizes more on overestimation. However, considering multiquantile, the prediction may suffer from crossing quantile problems that occur when all quantiles are estimated independently [26]. The problem was alleviated using a joint model that considers prediction over multiple quantiles. Also, reducing quantile loss leads to the reduction in the 1-Wasserstein distance between the target and the prediction [27].

C. Scaled Exponential Linear Unit

The rectified linear unit (ReLU) is the most commonly used activation function in DRL to avoid vanishing gradient problems. Though ReLU reduces the computation complexity and training time, it can sparse the NN due to the dead neuron problem. Leaky ReLU solves this problem but has exploding gradient problem, needs an alpha value, and becomes linear after differentiation. The SELU activation function solves all the problems and does a kind of internal normalization by converting the NN into a self-organizing NN [11]. Hence, it leads to robust learning

$$\text{SELU}(x) = \lambda \begin{cases} x, & \text{if } x > 0 \\ \alpha e^x - \alpha, & \text{if } x \leq 0 \end{cases} \quad (1)$$

where x is the input to the function, and α and λ are predetermined constants having values ≈ 1.67326 and ≈ 1.0507 , respectively.

D. Ensemble Policies

The concept of learning ensemble policies was used in different ways to get a more generalized and robust policy. The ensemble policies were learned with ensemble critics, and the best policy was selected using majority voting [28]. The actor and critic ensembles were also used to avoid dooming actions [17]. The actor ensemble was used in actor ensemble algorithm (ACE) to find the global maxima for the state, action value function in option-critic framework [16]. Another actor-critic approach, ac-Teach, considers an ensemble of teachers where the actor takes advice from multiple teachers [29] and uses Thompson sampling to choose a policy. The ensemble policies were also learned by learning diverse policies through parameter perturbation at regular intervals [15].

III. PROPOSED APPROACH

This section highlights how the aforementioned components are integrated with the actor-critic approach SAC to increase the robustness and generalization capabilities of the actor. We call the proposed approach QC_SANE as it uses quantile regression to train the critic, and the spikes of SNN govern the actor's actions.

QC_SANE: The proposed approach (QC_SANE)¹ is based on the "follow the winner" idea where the actor is trained indirectly through

¹https://github.com/surbhi1944/QC_SANE.git

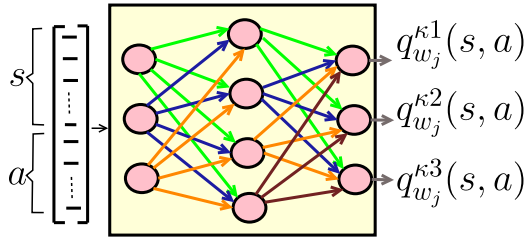


Fig. 1. Quantile critic with three quantiles κ_1 , κ_2 , and κ_3 .

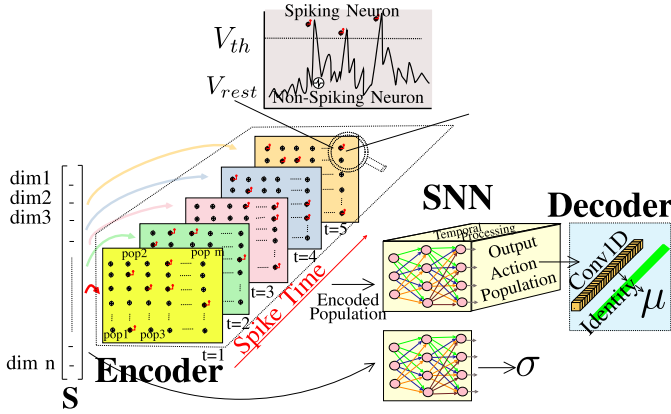


Fig. 2. Population coded spiking actor.

the agents. The actor interacts with the environment, collects the data, and sends the experienced data to all the agents (ensemble of actors) for training. After a certain amount of training, the actor assesses all the agents and decides to copy the winner of that round of training (strategy to select from the ensemble). The further interaction of the actor with the environment is with those fully adapted parameters of the best performer. In QC_SANE, the actor does not bond itself with the best agent of the initial round. Instead, it switches itself with the best performing agent in every round.

A. Preliminary

Our formulation of the Markov decision process (MDP) for continuous control task considers the finite horizon setting, where the actor explores a state space \mathcal{S} by performing actions from action space \mathcal{A} . Actor's action $a \in \mathcal{A}$ on a state $s \in \mathcal{S}$ arrives him in another state $s' \in \mathcal{S}$ based on the environment dynamics $p: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^x$ (where x : dimension of the state vector), with a reward feedback $r: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^1$ that is utilized by actor for learning using the off-policy DRL approach and an indicator d to show the episode termination.

B. Training QC_SANE

The training of QC_SANE involves the minimization of agent's Kullback-Leibler (KL) divergence loss and critics loss. QC_SANE uses an entropy-based SAC algorithm as the core, which considers the mean square loss at the critic. The proposed approach considers quantile loss. SELU nonlinearity is used at all layers except at the output layer as it resembles robust training. The q -value predictions of j th critic with parameters w_j are multiple quantiles κ_1 , κ_2 , and κ_3 , where each quantile prediction focuses on each feature of the state " s " and action " a " vector, as shown in Fig. 1. For all quantiles, we used the same NN, as approached in [30]. Since each quantile is estimated from the same NN, it avoids the crossing quantile problem. Critic

estimates its prediction loss $L_Q(w_j)$ using the following equations:

$$\delta_Q(w_j) = \mathbb{E}_{(s,a) \sim \mathcal{D}} \left[\mathbb{E}_{s' \sim p, a' \sim \pi_\theta} \left[r + \gamma \left\{ \min_{\bar{w}_j} \frac{1}{c} \sum_{k=1}^c Q_{\bar{w}_j}^{\kappa_k}(s', \pi_\theta(a'|s')) - \alpha_e \log(\pi_\theta(a'|s')) \right\} \right] - \frac{1}{c} \sum_{k=1}^c Q_{w_j}^{\kappa_k}(s, a) \right] \quad (2)$$

$$L_Q(w_j) = \sum_{k=1}^c \max(\kappa_k * \delta_Q(w_j), (\kappa_k - 1) * \delta_Q(w_j)) \quad (3)$$

where \bar{w}_j shows the critic's target network parameters and j shows the index of the considered critic who evaluates the quality of the action selected by the actor on the state. Our experiments focus on three quantiles ($c = 3$) and two critics.

Actor's actions are based on the learning of their ensemble agents that learn different policies. In some of the regions of state space, one agent's policy may outperform, while, in other regions, its policy underperforms other agent's policies. The learning at these ensemble agents happens by minimizing the loss

$$L_\pi(\theta_i) = \mathbb{E}_{s \sim \mathcal{D}} \left[\mathbb{E}_{a \sim \pi_{\theta_i}} \left[\alpha_e * \log(\pi_{\theta_i}(a|s)) - \min_j \frac{1}{c} \sum_{k=1}^c Q_{w_j}^{\kappa_k}(s, \pi_{\theta_i}(a|s)) \right] \right] \quad (4)$$

where π_{θ_i} represents the policy of the i th agent and θ_i shows its parameters. Each agent's policy network is a SNN with the population encoded states, actions, a decoder (Fig. 2), and an LIF model. The procedure of spike generation, encoding, and decoding is followed from [10] where each dimension of the state vector is encoded into the activity of neuron populations. Based on these populations, SNN (linear model) produces a population of action that is decoded for mean (μ) estimation of actual action using a 1-D convolution layer. The standard deviation (σ) is measured from a normal nonlinear NN with SELU activation. These μ and σ are used in the Gaussian distribution for the prediction of action

$$\mathcal{N}(\mu, \sigma) = \mu + \mathcal{N}(0, 1) * \sigma. \quad (5)$$

Fig. 3 shows the block diagram of the training of QC_SANE that uses three agents, one actor, and two critic networks. The target network of the critic that is used for the target estimation is updated from the critic's networks.

C. Searching Robust Policy Using Ensemble of Population

The proposed ensemble approach idea is different from that is used in evolutionary RL research where the population of agents evolves, reproduced, and is abolished over time based on their performance. The proposed approach lets the agents of the ensemble seek out the better policy till the end, which lets them become robust by considering most of the explored regions of the state space, learn catastrophic scenario, and find all the trajectories leading to a better endpoint.

D. Theoretical Convergence of Quantile-Based Evaluation Under Soft Policy Iteration

The proposed approach is based on SAC that is a practical approach derived from the theoretical convergence property of entropy augmented soft policy iteration (SPI) approach. The policy evaluation

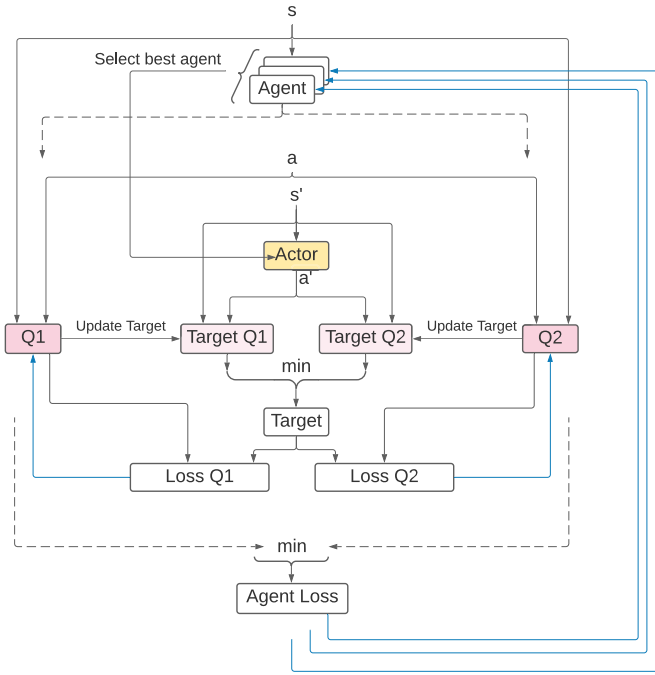


Fig. 3. Block diagram of QC_SANE where black and blue lines show forward and backward propagations, respectively. Dashed lines show forward pass with another input.

step of SPI uses the following operator:

$$\mathcal{T}^\pi Q(s, a) \triangleq r(s, a) + \gamma \mathbb{E}_{s' \sim p} [\mathbb{E}_{a' \sim \pi} [Q(s', a') - \log(\pi(a' | s'))]] \quad (6)$$

and has shown the convergence of this Bellman operator. The critic considered in QC_SANE predicts the quantile values. Hence, for the evaluation of policy π , the Bellman operator of (6) is updated

$$\mathcal{T}^\pi Q(s, a) \triangleq r(s, a) + \gamma \mathbb{E}_{s' \sim p} [\mathbb{E}_{a' \sim \pi} [\mathbb{E}_{\kappa_k \sim \kappa} Q^{\kappa_k}(s', a') - \log(\pi(a' | s'))]]. \quad (7)$$

Lemma 1: Consider the updated Bellman operator of (7) with $Q^0 : S \times \mathcal{A} \rightarrow \mathbb{R}^{|\kappa|}$, $|\mathcal{A}| < \infty$, and $Q^{i+1} = \mathcal{T}^\pi Q^i$. Then, \mathcal{T}^π is still a contraction map, and the sequence Q^i will converge as $i \rightarrow \infty$.

Proof: Redefine the operator by augmenting the entropy function \mathcal{H} in the reward

$$r_\pi(s, a) \triangleq r(s, a) + \mathbb{E}_{s' \sim p} [\mathcal{H}(\pi(\cdot | s))].$$

Hence, after rewriting the operator

$$\mathcal{T}^\pi Q(s, a) \triangleq r_\pi(s, a) + \gamma \mathbb{E}_{s' \sim p} [\mathbb{E}_{a' \sim \pi} [\mathbb{E}_{\kappa_k \sim \kappa} Q^{\kappa_k}(s', a')]].$$

To prove that \mathcal{T}^π is a contraction map, let $Q1(s, a)$ and $Q2(s, a)$ be two state action value functions, and the norm on the Q -value is defined as $\|Q1 - Q2\| \triangleq \|Q1 - Q2\|_\infty$. Let $Q1(s, a) > Q2(s, a)$

$$\begin{aligned} & \|\mathcal{T}^\pi Q1(s, a) - \mathcal{T}^\pi Q2(s, a)\| \\ &= |r_\pi + \gamma \mathbb{E}_{s' \sim p, a' \sim \pi} \mathbb{E}_{\kappa_k \sim \kappa} Q1^{\kappa_k}(s', a') \\ &\quad - r_\pi - \gamma \mathbb{E}_{s' \sim p, a' \sim \pi} \mathbb{E}_{\kappa_k \sim \kappa} Q2^{\kappa_k}(s', a')|_\infty \\ &= \gamma |\mathbb{E}_{s' \sim p, a' \sim \pi} \mathbb{E}_{\kappa_k \sim \kappa} Q1^{\kappa_k}(s', a') - \mathbb{E}_{s' \sim p, a' \sim \pi} \mathbb{E}_{\kappa_k \sim \kappa} Q2^{\kappa_k}(s', a')|_\infty \\ &\leq \gamma |\mathbb{E}_{s' \sim p, a' \sim \pi} \mathbb{E}_{\kappa_k \sim \kappa} [Q1^{\kappa_k}(s', a') - Q2^{\kappa_k}(s', a')]|_\infty \\ &\leq \gamma \|(Q1^{\kappa_k}(s', a') - Q2^{\kappa_k}(s', a'))\| \\ &\|\mathcal{T}^\pi Q1 - \mathcal{T}^\pi Q2\| \\ &\leq \gamma \|Q1 - Q2\|. \end{aligned}$$

Algorithm 1 QC_SANE

Input: Initialize:
SNN based ensemble actor parameters θ_i :
Set encoder μ, σ for I/P population
SNN parameters
Decoder parameters
Quantile levels $\kappa = \{\kappa_1, \kappa_2, \dots, \kappa_c\}$, where κ_k shows the chosen quantile
Quantile Critic parameters w_j
where $i \in [1, n]$, $j \in [1, m]$, $k \in [1, c]$

- 1 $s \leftarrow s_0$
- 2 **for** $t = 0, 1, 2, \dots, T$ **do**
- 3 **if** ($t > T_{ss}$) **then**
- 4 $a \leftarrow \pi_\theta(a|s)$
- 5 **else**
- 6 $a \leftarrow$ randomly sample action
- 7 **end**
- 8 $r, s', d \leftarrow$ Perform a in s
- 9 $\mathcal{D} \leftarrow \mathcal{D} \cup (s, a, r, s', d)$
- 10 $s = s'$
- 11 **if** d **then**
- 12 reset state s
- 13 **end**
- 14 **if** $t \geq u_a$ and $t \% u_e == 0$ **then**
- 15 **for** $u = 0, 1, 2, \dots, u_e$ **do**
- 16 Sample a batch (s, a, r, s', d) from \mathcal{D}
- 17 $w_j \leftarrow w_j - \lambda_Q \nabla_{w_j} L_Q(w_j)$
// Update critic parameters w_j
- 18 $\theta_i \leftarrow \theta_i - \lambda_\pi \nabla_{\theta_i} L_\pi(\theta_i)$
// Update agents (ensemble actor) parameters θ_i
- 19 $\bar{w}_j = \tau w_j + (1 - \tau) \bar{w}_j$
// Update target parameters
- 20 **end**
- 21 **end**
- 22 **if** $(t + 1) \% t_{test} == 0$ **then**
- 23 $\theta = \arg \max_{\theta_i} R(\pi_{\theta_i})$ // Update actor parameters θ
using agent getting high test return
- 24 **end**
- 25 **end**

TABLE I
VALUES OF EXPERIMENTAL PARAMETERS

Variable	Value	Variable	Value
No. of epoch	100	Actor(μ) learning rate λ_π	$1e^{-4}$
Spike time	5	Actor(σ) learning rate	$3e^{-4}$
τ	0.005	Critic learning rate λ_Q	$3e^{-4}$
Start steps T_{ss}	$10e^3$	Update every(u_e)	50
Encoder μ	(-3,3)	Encoder population dimension	10
Encoder σ	$\sqrt{0.15}$	Decoder population dimension	10
γ	0.99	Size of Replay buffer	$1e^6$
α_e	0.2	Steps per epoch or t_{test}	$10e^3$
No. of layers	2	No. of hidden units	256
c	3	Update after(u_a)	1000
κ	{0.1,0.5,0.9}	Test episode n_e	10
n	3	m	2

Similarly, we can prove by considering $Q1(s, a) < Q2(s, a)$ that

$$\|\mathcal{T}^\pi Q2(s, a) - \mathcal{T}^\pi Q1(s, a)\| \leq \gamma \|Q2 - Q1\|.$$

Hence, \mathcal{T}^π is a contraction map, and we can apply the standard convergence results for policy evaluation using [5]. \square

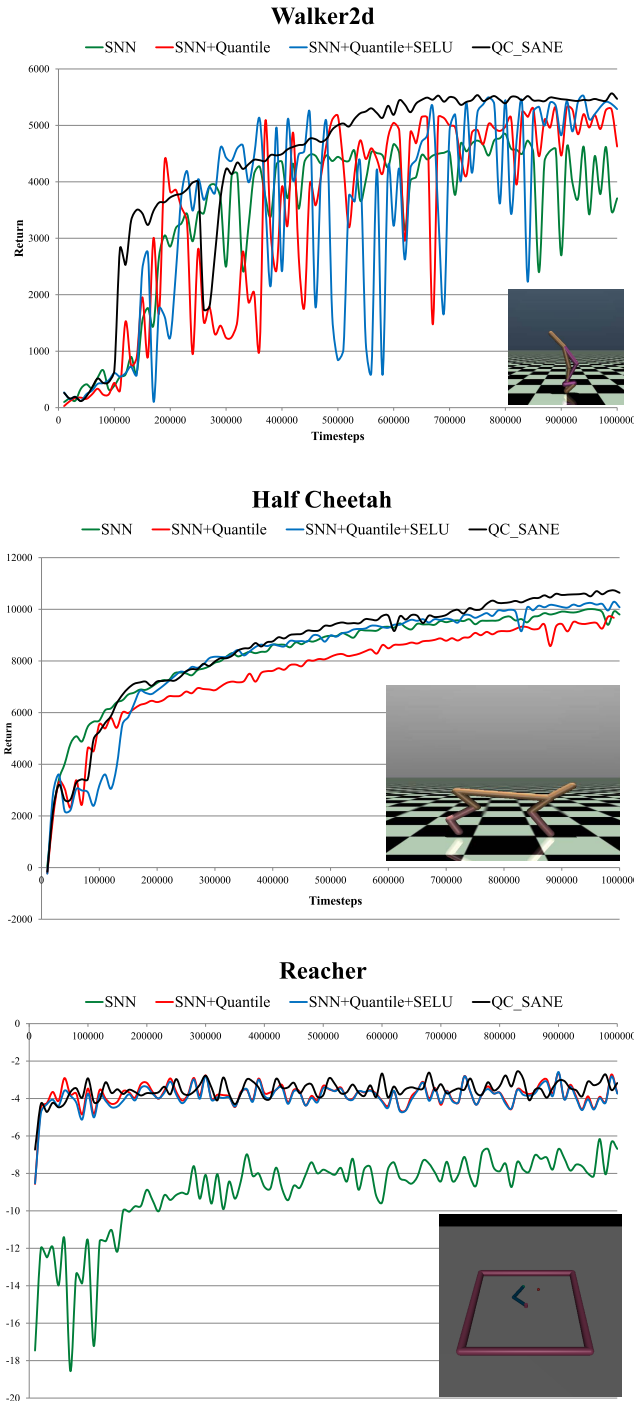


Fig. 4. Training time performance graph of population coded spiking actor network (SNN) and the proposed robust approach QC_SANE derived in ablation manner.

Based on the theoretical convergence, we derived the practical approximated version of quantile-based evaluation for critic following SAC. Algorithm 1 details the training steps of QC_SANE. In that, after every t_{test} , the policy of all the agents of actors π_{θ_i} is assessed by taking the mean of return of n_e episodes as

$$R(\pi_{\theta_i}) = \sum_t r_t(s_t, \pi_{\theta_i}(s_t)). \quad (8)$$

Then, actor is updated using the best performing agent.

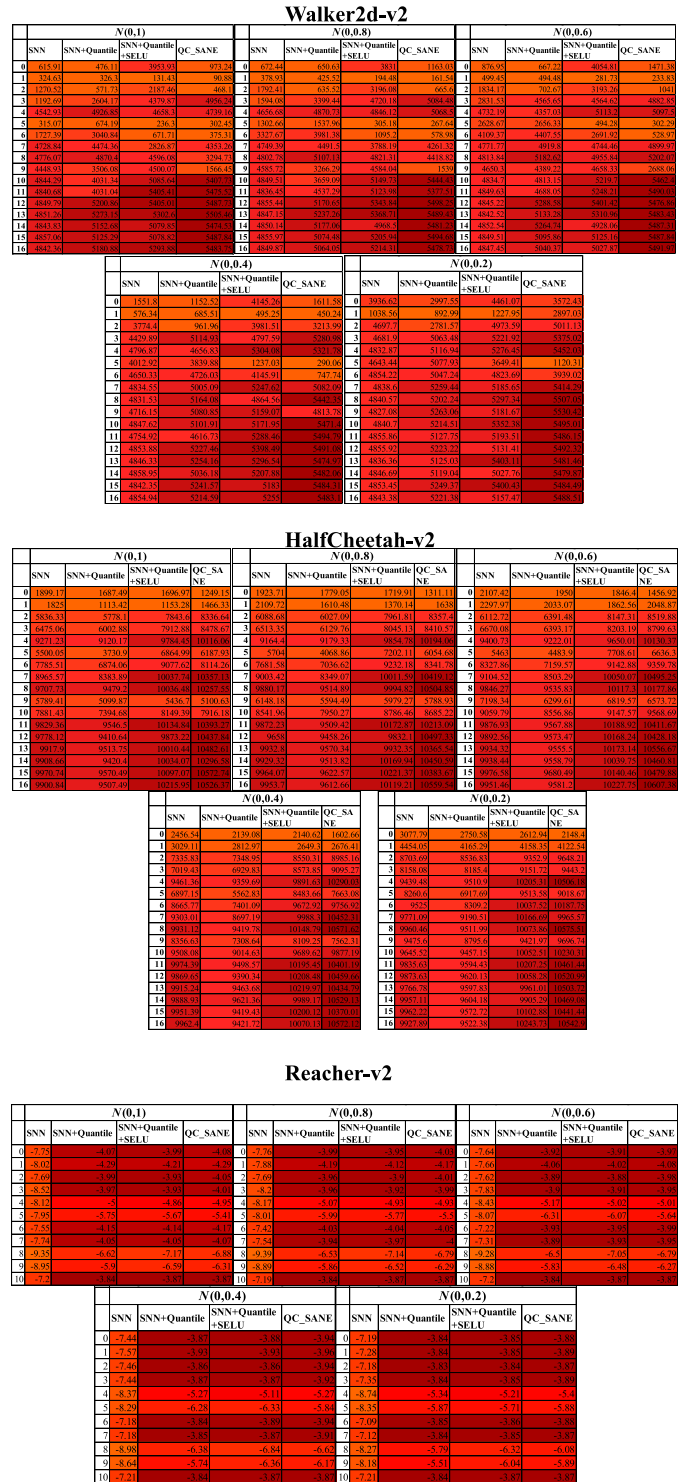


Fig. 5. Heatmap of the test time performance of the best model saved during training.

IV. EXPERIMENTS WITH RESULT AND DISCUSSION

Our experiments are done in the DRL benchmark environment OpenAI gym MuJoCo that models the dynamics of differently structure robot bodies. The environment models high-dimensional state and action spaces. Table I shows the values of the parameters used in the experiments.

The graphs for each algorithm in Fig. 4 are drawn by taking the mean of return of ten episodes at the end of every epoch. Due to the

TABLE II
SENSITIVITY ANALYSIS (μ AND σ) OF QC_SANE
FOR QUANTILES IN WALKER2D

Dim	$\mathcal{N}(0,1)$	$\mathcal{N}(0,0.8)$	$\mathcal{N}(0,0.6)$	$\mathcal{N}(0,0.4)$	$\mathcal{N}(0,0.2)$
0	(500, 415)	(575, 518)	(688, 683)	(854, 696)	(2182, 1453)
1	(346, 265)	(361, 229)	(457, 236)	(705, 239)	(3533, 703)
2	(1432, 1232)	(1542, 983)	(2264, 1240)	(3788, 525)	(5121, 102)
3	(2617, 2143)	(2858, 2032)	(3231, 1631)	(4126, 1001)	(5181, 180)
4	(4921, 184)	(5103, 33)	(5155, 52)	(5268, 56)	(5354, 101)
5	(290, 76)	(291, 77)	(328, 49)	(536, 256)	(3249, 1846)
6	(1477, 1946)	(1609, 1995)	(1937, 2255)	(2700, 1983)	(4741, 694)
7	(4247, 664)	(4434, 483)	(4720, 391)	(5077, 123)	(5314, 111)
8	(3388, 815)	(4327, 547)	(5149, 151)	(5381, 65)	(5410, 103)
9	(2290, 1449)	(2425, 1232)	(3495, 1171)	(4885, 368)	(5416, 107)
10	(5357, 64)	(5369, 84)	(5377, 98)	(5377, 108)	(5384, 126)
11	(5374, 104)	(5347, 65)	(5384, 114)	(5385, 124)	(5380, 121)
12	(5380, 113)	(5382, 122)	(5376, 112)	(5384, 121)	(5381, 121)
13	(5386, 129)	(5379, 121)	(5380, 116)	(5378, 115)	(5377, 120)
14	(5387, 96)	(5390, 102)	(5386, 115)	(5384, 110)	(5382, 116)
15	(5386, 120)	(5386, 121)	(5381, 121)	(5381, 121)	(5382, 121)
16	(5382, 117)	(5381, 117)	(5382, 124)	(5382, 115)	(5384, 121)

TABLE III
SENSITIVITY ANALYSIS (μ AND σ) OF QC_SANE
FOR AGENTS OF ACTOR IN WALKER2D

Dim	$\mathcal{N}(0,1)$	$\mathcal{N}(0,0.8)$	$\mathcal{N}(0,0.6)$	$\mathcal{N}(0,0.4)$	$\mathcal{N}(0,0.2)$
0	(2228, 1157)	(2160, 949)	(2227, 662)	(2599, 863)	(3410, 158)
1	(291, 181)	(373, 199)	(496, 240)	(794, 341)	(3476, 540)
2	(1659, 1061)	(2108, 1332)	(2520, 1283)	(4037, 721)	(5114, 118)
3	(4899, 126)	(4946, 150)	(4924, 92)	(5124, 174)	(5261, 147)
4	(4205, 780)	(4623, 581)	(4899, 306)	(5129, 220)	(5311, 167)
5	(766, 701)	(929, 973)	(1268, 1286)	(2156, 1776)	(3605, 2152)
6	(2441, 2156)	(2695, 2059)	(2740, 2163)	(3173, 2174)	(4729, 684)
7	(4473, 235)	(4373, 223)	(4758, 174)	(5044, 64)	(5274, 162)
8	(3951, 570)	(4708, 251)	(5226, 42)	(5387, 111)	(5400, 152)
9	(2412, 793)	(2828, 1202)	(3424, 682)	(4886, 115)	(5406, 164)
10	(5319, 163)	(5343, 162)	(5355, 157)	(5365, 147)	(5371, 160)
11	(5364, 146)	(5328, 111)	(5367, 154)	(5367, 158)	(5365, 154)
12	(5378, 144)	(5379, 149)	(5366, 151)	(5373, 157)	(5371, 159)
13	(5378, 161)	(5374, 154)	(5369, 154)	(5368, 148)	(5366, 158)
14	(5356, 165)	(5362, 158)	(5368, 157)	(5362, 163)	(5365, 156)
15	(5374, 159)	(5375, 162)	(5373, 158)	(5371, 158)	(5367, 159)
16	(5362, 162)	(5363, 153)	(5370, 161)	(5370, 155)	(5366, 156)

high computation budget and more training time, we have considered a single seed value of 100 for plotting those graphs, but the heatmap results of Fig. 5 are drawn by running five episodes for ten different seeds for each dimension, and their average is considered. The results at the training time for QC_SANE are either showing better or competitive performance than other experiments. The study is done in an ablation manner. The performance of each add-on is compared with the population coded spiking actor network (PopSAN) [10] (named SNN in graphs).

The training graphs of the Walker2d environment for each approach are showing more oscillating performance, while QC_SANE results are more stable and show less variability. For the HalfCheetah environment, though normalized networks using SELU have shown degraded performance, the overall performance using QC_SANE started outperforming other approaches from the middle of the training. In the Reacher environment, all approaches are performing similar and outperforming SNN.

Fig. 5 shows the heatmap of the performance under the Gaussian noise in the corresponding dimension (For Walker2d: 17, HalfCheetah:17, and Reacher: 11) of the state vector. The color shades show the robustness to noise. More darkness reflects more robustness, and lightness shows less robustness. The heatmap of the Reacher environment shows the robustness of all approaches except SNN. For Walker environment heatmap of QC_SANE shows more robust points. In the HalfCheetah environment, QC_SANE is again outperforming others.

For the QC_SANE, the number of agents n of actor and the number of quantiles c of critic are the two parameters. We have also done

sensitivity analysis (in Walker2d environment) to know the variance in the results with the change in the values of these parameters. To analyze the sensitivity for number of quantiles c , we kept the number of agents fixed ($n = 3$) and vary the parameter c such that, for $c = 2$, $\kappa = \{0.1, 0.9\}$; for $c = 3$, $\kappa = \{0.1, 0.5, 0.9\}$; and for $c = 4$, $\kappa = \{0.1, 0.9, 0.2, 0.8\}$. Table II shows the mean and standard deviation in the results for different quantile values. The results reflect less variance for higher dimensional values, and results on lower dimensional values show comparatively more variance.

Table III shows the sensitivity analysis for the agents. For this, we kept the number of quantiles fixed $c = 3$ and $\kappa = \{0.1, 0.5, 0.9\}$ and varied the number of agents to 2, 3, and 4. Here also the same trend is observed: lower dimensional values are having comparatively more variance than the higher dimensional values. Hence, the lower dimensions are more sensitive to the change in the number of quantiles and agents.

V. CONCLUSION AND FUTURE WORK

This work proposes an approach using quantile loss function, an ensemble of SNN-based actors with population-coded input-output, and normalized NN. The proposed approach is robust. As it uses an SNN, it is also friendly to advanced hardware that works in an energy-efficient manner. The demonstrated results show improved performance. The limitation of QC_SANE is the computation requirement that can be tackled in future work. Extending SNN-based QC_SANE with a truncated mixture of continuous distributional quantile critics to further increase the performance is relegated as part of future work.

REFERENCES

- [1] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
- [2] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
- [3] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q -learning," in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 1–7.
- [4] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.
- [5] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.
- [6] S. Gupta, G. Singal, and D. Garg, "Deep reinforcement learning techniques in diversified domains: A survey," *Arch. Comput. Methods Eng.*, vol. 28, pp. 4715–4754, Feb. 2021.
- [7] X. Ma, L. Xia, Z. Zhou, J. Yang, and Q. Zhao, "DSAC: Distributional soft actor critic for risk-sensitive reinforcement learning," in *Proc. Reinforcement Learn. Real Life Workshop ICML*, 2020, pp. 1–18.
- [8] K. Ota, T. Oiki, D. Jha, T. Mariyama, and D. Nikovski, "Can increasing input dimensionality improve deep reinforcement learning?" in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7424–7433.
- [9] D. Patel, H. Hazan, D. J. Saunders, H. T. Siegelmann, and R. Kozma, "Improved robustness of reinforcement learning policies upon conversion to spiking neuronal network platforms applied to atari breakout game," *Neural Netw.*, vol. 120, pp. 108–115, Dec. 2019.
- [10] G. Tang, N. Kumar, R. Yoo, and K. P. Michmizos, "Deep reinforcement learning with population-coded spiking neural network for continuous control," 2020, *arXiv:2010.09635*.
- [11] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," 2017, *arXiv:1706.02515*.
- [12] N. Tagasovska and D. Lopez-Paz, "Single-model uncertainties for deep learning," 2018, *arXiv:1811.00908*.
- [13] A. Kuznetsov, P. Shvechikov, A. Grishin, and D. Vetrov, "Controlling overestimation bias with truncated mixture of continuous distributional quantile critics," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5556–5566.
- [14] Y. Chung, W. Neiswanger, I. Char, and J. Schneider, "Beyond pinball loss: Quantile methods for calibrated uncertainty quantification," 2020, *arXiv:2011.09588*.

- [15] R. Saphal, B. Ravindran, D. Mudigere, S. Avancha, and B. Kaul, "SEERL: Sample efficient ensemble reinforcement learning," 2020, *arXiv:2001.05209*.
- [16] S. Zhang and H. Yao, "ACE: An actor ensemble algorithm for continuous control with tree search," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 5789–5796.
- [17] Z. Huang, S. Zhou, B. Zhuang, and X. Zhou, "Learning to run with actor-critic ensemble," 2017, *arXiv:1712.08987*.
- [18] M. Hessel *et al.*, "Rainbow: Combining improvements in deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.
- [19] A. Hill *et al.* (2018). *Stable Baselines*. [Online]. Available: <https://github.com/hill-a/stable-baselines>
- [20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018, *arXiv:1801.01290*.
- [21] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2014, pp. 387–395.
- [22] I. M. Comsa, K. Potempa, L. Versari, T. Fischbacher, A. Gesmundo, and J. Alakuijala, "Temporal coding in spiking neural networks with alpha synaptic function," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8529–8533.
- [23] J. C. V. Tieck, K. Secker, J. Kaiser, A. Roennau, and R. Dillmann, "Soft-grasping with an anthropomorphic robotic hand using spiking neurons," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2894–2901, Apr. 2021.
- [24] N. Rathi, G. Srinivasan, P. Panda, and K. Roy, "Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation," 2020, *arXiv:2005.01807*.
- [25] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, 1997.
- [26] I. Takeuchi *et al.*, *Nonparametric Quantile Estimation*. Cambridge, MA, USA: MIT Press, 2006.
- [27] O. Richter and R. Wattenhofer, "Learning policies through quantile regression," 2019, *arXiv:1906.11941*.
- [28] A. Hans and S. Udluft, "Ensembles of neural networks for robust reinforcement learning," in *Proc. 9th Int. Conf. Mach. Learn. Appl.*, Dec. 2010, pp. 401–406.
- [29] A. Kurenkov, A. Mandlekar, R. Martin-Martin, S. Savarese, and A. Garg, "AC-Teach: A Bayesian actor-critic method for policy learning with an ensemble of suboptimal teachers," 2019, *arXiv:1909.04121*.
- [30] H. Akrami, A. A. Joshi, S. Aydore, and R. M. Leahy, "Addressing variance shrinkage in variational autoencoders using quantile regression," 2020, *arXiv:2010.09042*.